SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
September 1996

3. REPORT TYPE AND DATES COVERED
Technical

4. TITLE AND SUBTITLE
Course Development in Parallel and Distributed Database Systems

5. FUNDING NUMBERS
DAAH04-95-1-0250

6. AUTHOR(S)
Jamal R. Alsabbagh and Carolyn Winston

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Grambling State University
Department of Math. and Computer Science
Carver Hall, Room 137
Grambling, Louisiana 71245

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 34157.27-MA-ISA

11. SUPPLEMENTARY NOTES
This is work in progress. It is part of a project titled "Integrating Research Results from Parallel and Distributed Processing into the Computer Science Curriculum" being carried out in collaboration with Louisiana Tech University and is partially supported by NSF.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Parallel Database Management Systems (PDBMS) address the performance requirements of heavily loaded systems that handle very large databases. Early attempts at meeting the requirements during the 1980's concentrated on building special database hardware. That approach was useful but the building of customized hardware was expensive. In contrast, the emergence of inexpensive commodity processors and other hardware shifted the focus, during the 1990's, to one of building extensible parallel computers. Three distinct architectures have been reported in the literature; they are shared memory, shared disk, and shared nothing. This presentation, made at the Annual ADSRC Meeting on September 96, contrasts the three architectures.

14. SUBJECT TERMS
Parallel Database Systems

15. NUMBER IF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Enclosure 1

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Jgra

COURSE DEVELOPMENT
IN
PARALLEL AND DISTRIBUTED DATABASE MANAGEMENT SYSTEMS

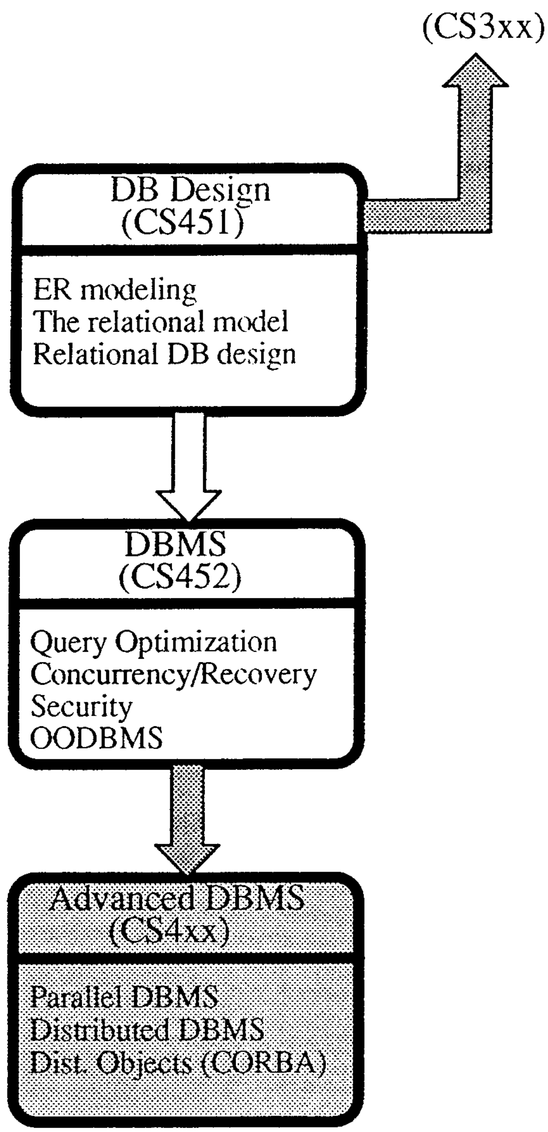
Jamal Alsabbagh and Carolyn Winston

ADSRC - Grambling State University

OUTLINE

- Context and Plan for This Work
- Overview of Parallel Database Systems

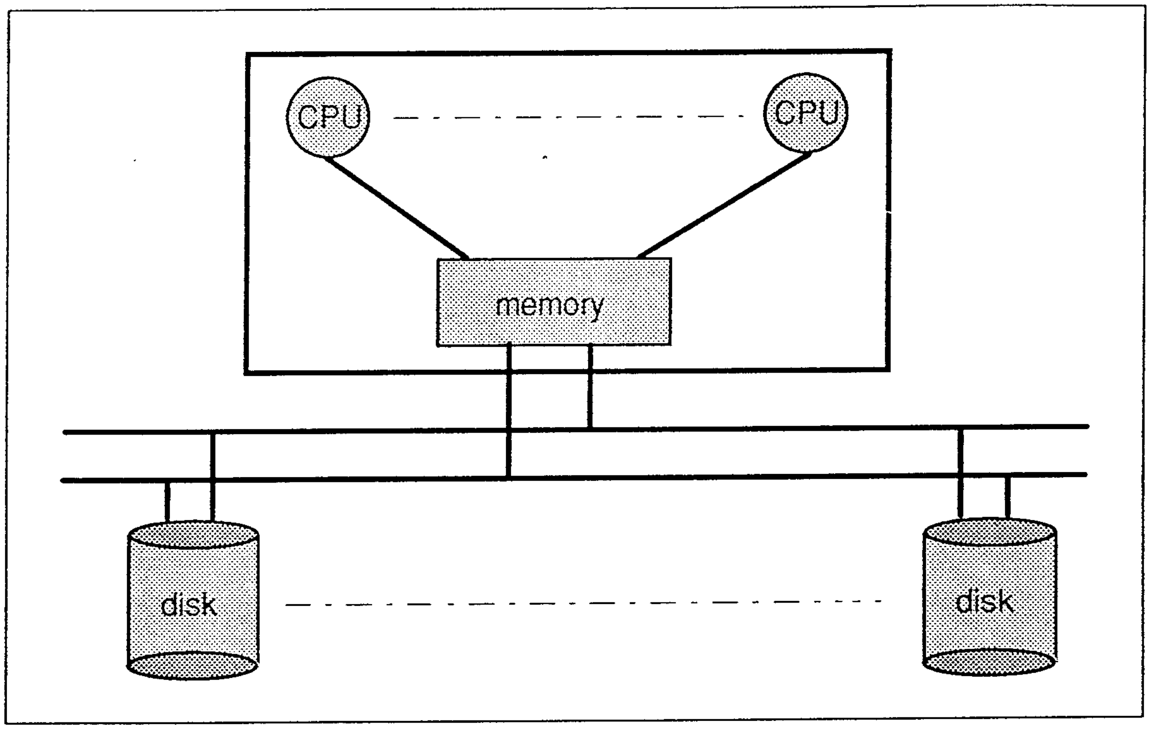
CURRENT AND PROPOSED DATABASE COURSES
(AT GSU)



Classification of Parallel Database Systems

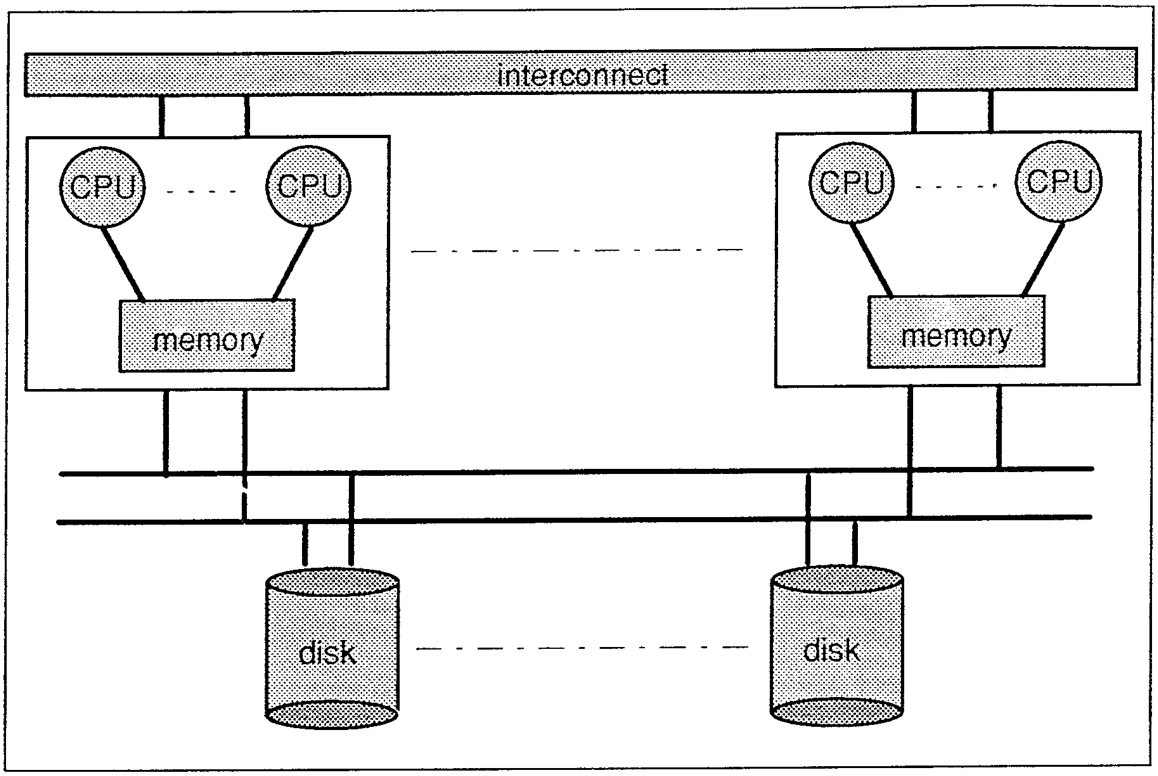
- Shared Memory (SM); also called Shared Everything (SE)
- Shared Disk (SD)
- Shared Nothing (SN)

Shared-Memory (SM) Parallel Database Systems



- All processors directly access, in a symmetrical fashion, all main memory and disks.
- In general, the operating system allocates processors to processes.
- Processors have local caches to reduce network traffic, but loading/flushing caches can degrade performance.
- Hardware-specific solutions are required to ensure coherence among the caches (e. g. processors continuously snoop the shared bus to see if their cached data is required elsewhere.)
- Typical Hardware: IBM 3090, IBM 370, Bull DBS8, Encore, Sequent Symmetry

Shared-Disk (SD) Parallel Database Systems

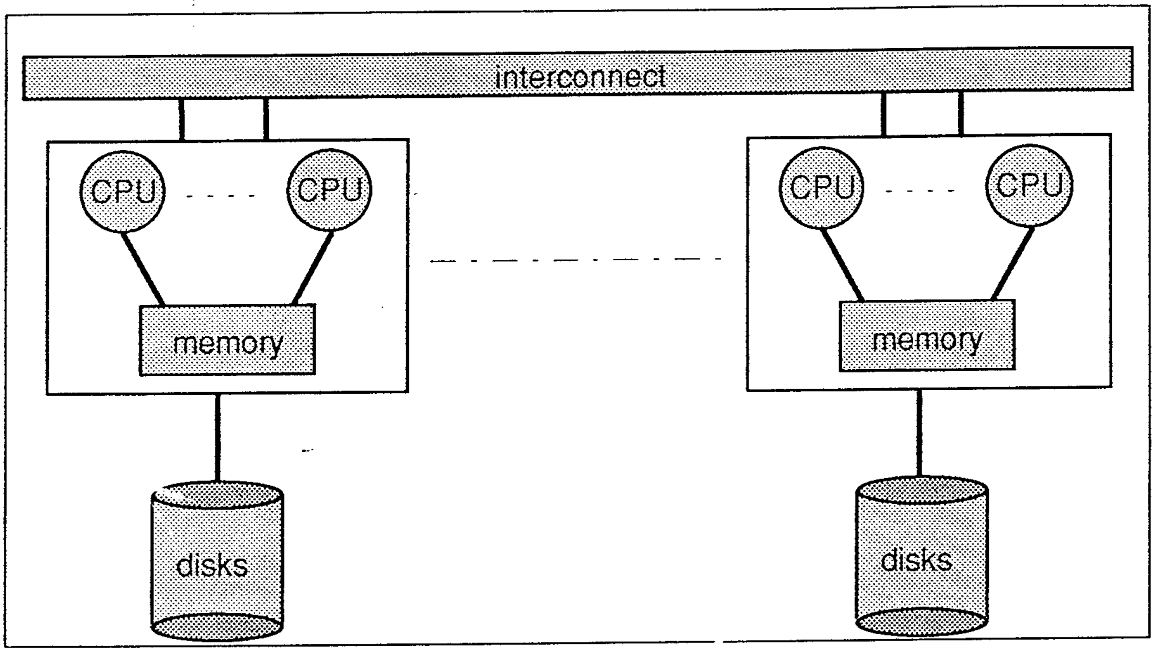


- Each node has its own memory and may itself be an SMP box.
- The nodes share the same disks logically (and may be physically too).
- System (or application) software must ensure the coherency among multiple copies of disk pages requested by different nodes.
- A query or update by a node requires it to:

  1. Transmit, to all other nodes, an intention to query/update the database.
  2. If the required page is currently being updated by any other node, then wait until it is released.
  3. Read or receive the required page.
  4. perform the query or update.

- Typical Hardware: DEC VM Cluster, SUN Sparc 1000 cluster

Shared-Nothing (SN) Parallel Database Systems



- Each node has its own local memory and its own local disks.
- The database is partitioned across the nodes, thus allowing I/O parallelization
- Each node acts as a server for its local data.
- An update by a processor requires:

  1. Transmit a request for update to the relevant server.
  2. The server performs the update, locally.
  3. The server acknowledges the success back to the requester.

- A query by a processor requires:

  1. Transmit the query to the relevant server.
  2. The server performs the query locally.
  3. The server sends the query result to the requester.

- Typical Hardware: AT&T 3600, IBM SP2, nCUBE, VAXcluster

Metrics for Evaluating the Three Architectures

	Metric	Explanation
1	**Price**	Using commodity hardware reduces system cost.
2	**Throughput**	Inter-query parallelism improves throughput.
3	**Response Time**	Intra-query parallelism improves response time.
4	**Speedup**	Ideally, twice the hardware should solve the problem in half the time.
5	**Scaleup**	Ideally, twice the hardware should solve twice the problem in the same time.
6	**Startup Cost**	Preparing a query for parallel execution is an overhead.
7	**Interference**	Processors slow each other when competing for shared resources.
8	**Load Balancing**	Ideally, all the processors should be working concurrently.
9	**Comm. Overheads**	Ideally, sub-problems of one problem should require least communication.
10	**Data Availability**	It is desirable to be able to tolerate failure of some nodes.
11	**Portability**	Porting centralized DBMS software should be relatively easy.